

# Funciones

## Funciones de MySQL

Mediante la sentencia SELECT es posible hacer una proyección de una tabla, seleccionando las columnas de las que queremos obtener datos.

Las expresiones SELECT no se limitan a nombres de columnas de tablas, pueden ser otras expresiones, incluso aunque no correspondan a ninguna tabla.

Es posible aplicar funciones sobre columnas de tablas, y usar esas columnas en expresiones para generar nuevas columnas.

Es posible asignar un alias a cualquiera de las expresiones select. Esto se puede hacer usando la palabra AS, aunque esta palabra es opcional.

### Sintaxis

```
SELECT Fecha, DATEDIFF( CURRENT_DATE(), Fecha ) Dias FROM Facturas;
```

### Sintaxis

```
SELECT ROUND(precio*1.21,2) as Iva FROM Facturas;
```

### Sintaxis

```
SELECT CONCAT('Hola ', 'Mundo') AS Saludo;
```

## Funciones de Agregado

Existen en SQL funciones que nos permiten contar registros, calcular sumas, promedios, obtener valores máximos y mínimos. Estas funciones se denominan **Funciones de Agrupamiento** porque operan sobre conjuntos de registros, no con datos individuales.

Tienen la característica de agrupar los resultados en un solo registro de salida.

### COUNT()

Retorna la cantidad de valores que contiene un campo especificado. Por ejemplo, si se quiere saber la cantidad de productos que hay en la tabla artículos:

### Sintaxis

```
SELECT COUNT (*) FROM Articulos;
```

También es posible combinarla con la cláusula WHERE. Por ejemplo si se quiere saber cuántos productos tienen por Nombre "iPhone" de en tabla artículos:

### Sintaxis

```
SELECT COUNT (*) FROM Articulos WHERE Nombre LIKE "%Pala%";
```

Tener en cuenta que **no debe haber espacio entre el nombre de la función y el paréntesis** pues puede confundirse con una referencia a una tabla o campo.

Es decir...

<code>SELECT COUNT(*) FROM Productos;</code>	Correcto :)
<code>SELECT COUNT (*) FROM Productos;</code>	Incorrecto :(

## SUM()

Retorna la suma de los valores que contiene el campo especificado. Por ejemplo, si se quiere saber el stock de productos que hay en la tabla artículos:

### Sintaxis

```
SELECT SUM(Stock) FROM Articulos;
```

## MIN()

Para averiguar el valor mínimo de un campo usamos las funciones **min()**. Ejemplo, se quiere saber cuál es el menor precio de todos los artículos:

### Sintaxis

```
SELECT MIN(Precio) FROM Articulos;
```

## MAX()

Para averiguar el valor máximo de un campo usamos las funciones **max()**. Ejemplo, se quiere saber cuál es el mayor precio de todos los artículos:

### Sintaxis

```
SELECT MAX(Precio) FROM Articulos;
```

## AVG()

Retorna el valor promedio de los valores del campo especificado. Por ejemplo, si se quiere saber el promedio del precio de los artículos:

### Sintaxis

```
SELECT AVG(Precio) FROM Articulos;
```

# Funciones de Agrupamiento Avanzado

## GROUP BY

La agrupación es un concepto básico de Base de Datos, La cláusula GROUP BY, como su traducción lo indica tiene como propósito agrupar información de acuerdo a un criterio en común. Por lo general se utiliza con funciones de agrupación o de agregación (COUNT, MIN, MAX, AVG, SUM).

ID	Nombre	Precio	Marca	Categoría	Presentacion	Stock	Disponible
1	iPhone 6	499.99	Apple	Smartphone	16GB	500	SI
2	iPad Pro	599.99	Apple	Smartphone	128GB	300	SI
3	Nexus 7	299.99	LG	Smartphone	32GB	250	NO
4	Galaxy S7	459.99	Samsung	Smartphone	64GB	200	SI
5	Impresora T23	489.99	Epson	Impresoras	Color	180	NO
6	Impresora T33	399	Epson	Impresoras	Color	200	NO
7	Lavarropa 7000	1679	LG	Lavarropas	Automático	100	SI
8	Camara Digital 760	649	Kodak	Fotografía	Sin detalle	150	NO
9	Notebook CQ40-300	2999	HP	Notebooks	Intel Core i3	100	SI

Suponiendo que se quiere obtener el número de productos existentes en la categoría "Smartphone", entonces habrá que agrupar los registros por cada Categoría.

La función GROUP BY permite hacer esto de manera automática a partir de tomar un valor o dato común.

## Sintaxis

```
SELECT Categoria, SUM(Stock) FROM Productos GROUP BY Categoria;
```

<b>Categoria</b>	<b>Stock</b>
<b>Smartphone</b>	<b>1250</b>
<b>Impresoras</b>	<b>380</b>
<b>Lavarropas</b>	<b>100</b>
<b>Fotografía</b>	<b>150</b>
<b>Notebooks</b>	<b>100</b>

El comportamiento de la función GROUP BY dependerá de la función de agrupación que se esté utilizando.

## HAVING

La cláusula HAVING permite hacer selecciones en situaciones en las que no es posible usar WHERE. Dado que se establece un criterio sobre un valor dado por una función de agrupamiento y no por valores de registros.

## Sintaxis

```
SELECT Categoria, SUM(Stock) FROM Productos GROUP BY Categoria  
HAVING SUM(Stock) > 250;
```

Categoria	Stock
Smartphone	800
Impresoras	380

## Lenguajes DDL y DML

### Lenguaje de Definición de Datos DDL

Es el conjunto de comandos que se usa para crear bases de datos y tablas, y para modificar sus estructuras, así como los permisos y privilegios.

Se caracteriza por involucrar los comandos necesarios para crear, modificar y eliminar una tabla; además de crear claves primarias, índices y restricciones.

CREATE	DROP
ALTER	TRUNCATE

### Lenguaje de Manipulación de Datos DML

Es el que se usa para modificar y obtener datos desde las bases de datos.

Se caracteriza por involucrar los comandos necesarios para hacer consultas, inserciones, modificaciones y eliminaciones.

INSERT	UPDATE
DELETE	SELECT

